Blackjack.py

Your assignment is to write a program that simulates one round of the card game Blackjack between two players, you and the dealer (the computer).

Basic Rules of Blackjack:

Blackjack is a card game played with a standard deck of 52 cards. Cards are ranked from high to low in the order Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2. There are four suits: Spades, Hearts, Diamonds, and Clubs. The suits are of equal value.

The goal in each round of Blackjack is to accumulate a higher point total than the dealer *without going over 21 points*. The point value of your hand at any given time is calculated by adding up the values of your individual cards. The cards 2 through 10 have value equal to their ranks (that is, a 2 counts 2 points, an 8 counts 8 points, etc. The Jack (J), Queen (Q), and King (K) are worth ten points each. The Ace (A) is worth either 1 point or 11 points: the player chooses which is more advantageous, and can change his/her mind at any time during his/her turn. The suits of the cards have no effect on the point value of a hand.

Examples:

- 4S KH = 14 (four, plus ten for the King)
- QS QD = 20 (each queen is worth ten points)
- 3H 4S AH 6D 2D 2S AS = 19 (there is no limit to the number of cards you have as long as your total doesn't exceed 21)
- AH 8D = 19 or 9 (19 if you choose 11 points for the ace, 9 if you choose 1 point)
- AC AD = 12 or 2 (both aces can't be 11 points since that would put you over 21 points)

At the start of a blackjack game, the player and the dealer each receive two cards. The players' cards are normally dealt face up, while the dealer has one face down (called the "hole card") and one face up. The player should never know the value of the dealer's hole card until the player completes his/her turn. After the initial cards have been dealt, the game begins. The player always goes first.

The best possible blackjack hand is an opening deal of an ace with any ten-point card. This is called a "blackjack", or a "natural" 21. A player dealt a blackjack immediately ends his/her turn, and wins unless the dealer has also been dealt a blackjack. If both the player and the dealer have a blackjack, the tie goes to the dealer.

The player can choose to keep his/her hand as it is ("stand" or "stay") or request more cards from the deck ("hit"), one at a time, until either the player judges that the hand is strong enough to win, and stands; or until the value of the hand exceeds 21, in which case the player immediately loses ("busts").

A common strategy is to count an Ace as 11 points at the start, in an attempt to get a blackjack or a favorable hand total. If the player does not like the result (such as busting), he/she can then choose to count the Ace as 1 point instead, and continue drawing cards.

Sample strategy for the examples above:

- 4S KH = 14: A total of 14 is relatively easy for the dealer to beat. If you "hit" (draw another card) on this hand, there is a risk of drawing an 8 or higher, "busting" and losing; however, drawing a 7 or less would improve your hand. Those odds, plus the fact that is is very likely that the dealer can do better than a 14, means you would probably hit rather than stay.
- QS QD = 20 (each queen is worth ten points). The only way to improve this hand would be to draw an ace to make the total exactly 21. Since the odds are extremely small this would happen, you should stay.
- 3H 4S AH 6D 2D 2S AS = 19 Although it's possible to improve this hand, the odds suggest you stay at 19.
- AH 8D = 19 or 9 (19 if you choose 11 points for the ace, 9 if you choose 1 point). Although it's possible to improve this hand, the odds suggest you stay at 19.
- AC AD = 12 or 2 (both aces can't be 11 points since that would put you over 21 points). 12 is too easy to beat. You should hit to see how close to 21 you can get with additional cards. If you go over 21, you have the option to count the ace as 2 instead, and either stay with your new total or hit again.

If the player hasn't "busted", it becomes the dealer's turn. The dealer reveals the facedown "hole card", and all play is done openly as the dealer tries to beat the player's hand. The dealer begins drawing cards one at a time until one of three things happens:

- If the dealer's hand exceeds 21 (i.e., the dealer "busts"), the player wins.
- If the dealer's hand is higher than or equal to the player's hand, the dealer wins. (Ties go to the dealer.)
- If the dealer's hand is still lower than the player's hand, the dealer draws another card.

For the purpose of this assighment, you can assume:

- The game is limited to two players, you and the dealer (the computer).
- You will play only one hand, using a complete deck of 52 shuffled cards. (Although it would be fun to play multiple hands, this would require you to manage a list of "used cards", and reshuffle the deck when you run out.)
- Since we're only playing one hand, no betting needs to take place, and you will not need to keep track of wins and losses.
- You do not need to implement any additional or optional rules of Blackjack, such as "splitting" and "five-card Charlies", that are not specified in this assignment.

To get started:

- Create a class `Card`.

  - When you create an instance of `Card`, you pass it a `Suit` ("C", "D", "H", or "S") and a `Rank` (2, 3, 4, 5, 6, 7, 8, 9, 10 J, Q, K, A). These values should be stored as instance variables for the new card object.
  - I recommend you also have an instance variable `value` that stores the point value of the card. You will find this very useful when you have to calculate the point value of a hand.
  - `Card` should have an `__str__` method that lets you print an individual card (such as "3S" for the three of spades).
- Create a class `Deck`.
  - When you create an instance of `Deck`, it should create a list of 52 `Card` objects and save it to an instance variable `cardList`.
  - It should have a `shuffle()` method that rearranges the cards in `cardList`. You can do this easily by importing the `random` package into Python, and using the`random.shuffle()` method. `random.shuffle(myList)` rearranges the elements of the list `myList` into a random order. Also, if you insert the statement `random.seed(#)` (where "#" is an integer) before calling "shuffle", you will always get the same shuffle. This is useful for debugging.
  - It should have a `dealOne()` method that removes the first card from your deck's `cardList`, and appends it to the hand of a specified player.
  - It should have an `__str__` method that lets you print out the entire deck in four rows of 13 cards (neatly aligned into columns) for debugging purposes. Note that the `__str__` method for `Deck` objects should take advantage of the `__str__` method for `Card` objects.
- Create a class `Player`.
  - When you create an instance of `Player`, it should have an instance variable `hand` that's set to an empty list, and an instance variable `handTotal` that's set to zero. These instance variables will be modified by the `Deck` class' `dealOne()` method.
  - It should have an `__str__` method that lets you print out the hand of a `Player`.

Your main program might look something like this:

```
def main()

    cardDeck = Deck()               # create a deck of 52 cards called 
"cardDeck"
    print(cardDeck)                 # print the deck so we can see that you 
built it correctly

    random.seed(50)                 # leave this in for grading purposes
    cardDeck.shuffle()              # shuffle the deck
    print(cardDeck)                 # print the deck so we can see that your 
shuffle worked

    dealer = Player()               # create the player:  you play for this 
Player
    opponent = Player()             # create the dealer:  the computer plays 
for this Player

    cardDeck.dealOne(opponent)      # face up
    cardDeck.dealOne(dealer)        # face down (the "hole" card)
    cardDeck.dealOne(opponent)      # face up
    cardDeck.dealOne(dealer)        # face up

    showHands(opponent,dealer)      # remember not to show face down cards

    opponentTurn(cardDeck,dealer,opponent)     # this is where half of the 
hard stuff is done
    dealerTurn(cardDeck,dealer,opponent)       # this is where the other half 
of the hard stuff is done

    print ("Game over.")
    print ("Final hands:")
    print ("   Dealer:   ", dealer)
    print ("   Opponent: ", opponent)

main()
```

Function `opponentTurn()` should prompt the user for actions ("hit" or "stand").
Both `opponentTurn()` and `dealerTurn()` should display the cards in play in an easy-to-read fashion.

Input and Output:

Your program should display the appropriate information to allow a player to make decisions during his/her turn, display updated hands after each card is drawn, and display what's happening during the dealer's turn. Finally, it should declare who wins each hand.

Your program should also display the shuffled deck at the start of the hand. Although in a real card game, the order of the cards would never be revealed in advance to the players, doing this will enable you to debug your code easily (that is, you can see what cards were dealt to ensure the hands have been correctly updated), and it will enable us to confirm that your program is handling the cards correctly during grading.

Here are two examples of expected output:

Sample Output 1: result with seed = 50

```
Initial deck:
  2C  3C  4C  5C  6C  7C  8C  9C 10C  JC  QC  KC  AC
  2D  3D  4D  5D  6D  7D  8D  9D 10D  JD  QD  KD  AD
  2H  3H  4H  5H  6H  7H  8H  9H 10H  JH  QH  KH  AH
  2S  3S  4S  5S  6S  7S  8S  9S 10S  JS  QS  KS  AS

Shuffled deck
  5D  7D  AH  9S 10S  5S  8H  KH  6C  KS  5C  3C  KC
  2S  AS 10C  KD  9C  2C  AD  4H  2H  QS  8S  9H  2D
  5H  8D  QC  3H  4C  AC  JS  4S  QH  8C  JD  JC  6S
  JH  3D  9D 10H  7C 10D  6H  7S  4D  3S  QD  6D  7H

Deck after dealing two cards each:
 10S  5S  8H  KH  6C  KS  5C  3C  KC  2S  AS 10C  KD
  9C  2C  AD  4H  2H  QS  8S  9H  2D  5H  8D  QC  3H
  4C  AC  JS  4S  QH  8C  JD  JC  6S  JH  3D  9D 10H
  7C 10D  6H  7S  4D  3S  QD  6D  7H



Dealer shows 9S faceup
You show AH faceup

You go first.

Assuming 11 points for an ace you were dealt for now.
You hold 5D AH  for a total of 16
1 (hit) or 2 (stay)? 1

Card dealt:  10S
Over 21.  switching an ace from 11 points to 1.
New total: 16

You hold 5D AH 10S  for a total of 16
1 (hit) or 2 (stay)? 1

Card dealt:  5S
21!  My turn. . .
Dealer's turn
Your hand:  5D  AH  10S  5S    for a total of 21
Dealer's hand:  7D  9S    for a total of 16

Dealer hits:  8H
New total: 24
```

```
Dealer has 24.  Dealer busts!  You win.

Game over.
Final hands:
   Dealer:    7D  9S  8H
   Opponent:  5D  AH  10S  5S
>>>
```

Sample Output 2: result with seed = 25

```
Initial deck:
  2C  3C  4C  5C  6C  7C  8C  9C 10C  JC  QC  KC  AC
  2D  3D  4D  5D  6D  7D  8D  9D 10D  JD  QD  KD  AD
  2H  3H  4H  5H  6H  7H  8H  9H 10H  JH  QH  KH  AH
  2S  3S  4S  5S  6S  7S  8S  9S 10S  JS  QS  KS  AS

Shuffled deck
  JH  AS  QD  4H  9S  QC  6C  KS  5S 10C  7H  JC  6D
  5H  3C  3D  7C  AH  AD  JS 10H  7D  4S  2S 10D  9H
  2H  5C  8S  4D  6S  JD  KC  9D  8H  AC  9C  KH  8C
  3H  QH 10S  7S  5D  4C  6H  3S  8D  2D  2C  QS  KD

Deck after dealing two cards each:
  9S  QC  6C  KS  5S 10C  7H  JC  6D  5H  3C  3D  7C
  AH  AD  JS 10H  7D  4S  2S 10D  9H  2H  5C  8S  4D
  6S  JD  KC  9D  8H  AC  9C  KH  8C  3H  QH 10S  7S
  5D  4C  6H  3S  8D  2D  2C  QS  KD



Dealer shows 4H faceup
You show QD faceup

You go first.

You hold JH QD  for a total of 20
1 (hit) or 2 (stay)? 2
Staying with 20

Dealer's turn
Your hand:  JH  QD    for a total of 20
Dealer's hand:  AS  4H    for a total of 15

Assuming 11 points for an ace I was dealt for now.
Dealer hits:  9S
New total: 24

Over 21.  switching an ace from 11 points to 1.
New total: 14

Dealer hits:  QC
New total: 24
```

```
Dealer has 24.  Dealer busts!  You win.

Game over.
Final hands:
   Dealer:    AS  4H  9S  QC
   Opponent:  JH  QD
>>>
```